Εργασία Αντικειμενοστραφής Προγραμματισμός ΙΙ
2022-2023

Νικόλαος Ιωάννης Καραπιπέρης <it22042>
Βασίλειος Τσιρμάκος <it22111>
Ηλίας Παναγόπουλος <it22083>

Στο πρώτο μέρος της εργασίας, δημιουργήσαμε 3 κλάσεις:

- **class Audio**: Η οποία περιέχει τρία attributes και τον default constructor. Επίσης, περιέχει ολές τις μεθόδους για την αναπαραγωγή mp3 files. Αναλυτικότερα, έχουμε την μέθοδο **playMp3UntilEnd,** η οποία δέχεται 2 ορίσματα, ένα **InputStream** και ένα **array** άπο **Strings.** Στην μεταβλήτη τύπου **InputStream**, περνάμε ως όρισμα το πρώτο στοιχείο του πίνακα **args**, το οποίο είναι και το πρώτο όρισμα που "δίνει" ο χρήστης. Τέλος, μέσω της **getter** μεθόδου της μεταβλητής **Player** αναπαράγουμε το mp3 file που έχει δώσει ως argument ο user. Η συγκεκριμένη μέθοδος περιέχει επίσης, κάποια println ώστε να ξέρει ο χρήστης ποιό τραγούδι μόλις ξεκίνησε και ποιό τελείωσε. Η κλάση **Audio** περιέχει επίσης, την μέθοδο **playMp3Loop**, η οποία δέχεται και αυτή 2 ορίσματα, ένα **InputStream** και ένα **array** άπο **Strings.** Η συγκεκριμένη μέθοδος κάνει ότι και η προηγούμενη με την εξαίρεση ότι, σε αυτή έχουμε δηλώσει μια boolean μεταβλητή να είναι ίση με true και έχουμε δημιουργήσει ενα **do while();** το οποίο αναπαράγει ένα τραγούδι απο την αρχή έως το τέλος και μολις τελειώσει το τραγούδι, το ξανά αναπαράγει απο την αρχή, μέχρις ότου ο χρήστης να τερματίσει το πρόγραμμα ή το **status** του **Player** να ισούται με **Player.status.ERROR.** Επίσης, έχουμε την μέθοδο **playM3uRandom**, η οποία δέχεται 3 ορίσματα, ένα **InputStream**, ένα **String** και ένα **M3uPlaylist**, το οποίο είναι αναφορά σε άλλη κλάση**.** Στην συγκεκριμένη μέθοδο, δημιουργούμε ένα αντικείμενο της κλάσης **M3uPlaylist** (θα εξηγήσουμε παρακάτω την συγκεκριμένη κλάση) και του περνάμε ως όρισμα το String args , το οποίο αντιστοιχεί στο **m3u file** που έδωσε ο χρηστης ως όρισμα. Στην συνέχεια, δημιουργούμε έναν πίνακα απο ακεραίους και θέτουμε το μεγεθός του ίσο με το μέγεθος του **m3u file**(Πόσα διαφορετικά κομμάτια περιέχει η λίστα), έπειτα θέτουμε τιμές στον πίνακα απο 0 έως το μέγεθος του **m3u file** και με την μέθοδο **shuffle '**ανακατεύουμε**'** με τυχαία σειρά τα στοιχεία του πίνακα. Αυτό είχε ως σκοπό την τυχαία αναπαραγώγη των τραγουδιών της λίστας και κάθε φορά να αναπαράγει κομμάτι, το οποίο δεν έχει αναπαραχθεί ακόμη. Στην συνέχεια, έχουμε την μέθοδο, **playM3uAllList**, η οποία δέχεται 3 ορίσματα, ένα **InputStream**, ένα **String** και ένα **M3uPlaylist.** Η συγκεκριμένη μέθοδος κάνει ότι και η προηγούμενη με την διαφορά ότι, αναπαράγουμε τα κομμάτια της λίστας με την σείρα το ένα μετά το άλλο. Επιπρόσθετα, έχουμε την μέθοδο **songsNames**, η οπόια βρίσκει τα ονόματα των τραγουδιών. Τέλος, έχουμε την μέθοδο **relativePaths,** μέσω της οποίας μπορούμε πλέον να αναπαράγουμε κομμάτια απο το **m3u file** τα οποία έχουν αναφορά σε **relative path.**
- **class M3uPlaylist**: Η συγκεκριμένη κλάση δημιουργήθηκε με σκόπο το **parsing** του **m3u file.** Περιέχει 2 **γνωρίσματα**, έναν **constructor** ο οποίος δέχεται ως input μια μεταβλητή τύπου **File** και κάνει τα εξής κάθε φορά που δημιουργείται ενα αντικείμενο της συγκεκριμένης κλάσης. Δημιουργεί ένα **arrayList,** με σκοπό να αποθηκευτούν εκεί τα mp3 files, δημιουργεί έναν **BufferedReader,** μέσω του οποίο μπορούμε να διαβάσουμε το αρχείο που θα δώσει ο χρηστης και μέσω της μεθόδου του **readLine()**, να διαβάσουμε το αρχείο γραμμή γραμμή, όπως και κάνουμε. Ελέγχουμε, αν η κάθε γραμμή του αρχείου ξεκινάει με '#', ώστε να απορρίψουμε τα τυχόν σχόλια που μπορεί να υπάρχουν στο αρχείο και θέτουμε ως όρισμα στην μέθοδο **addMp3**, την εκάστοτε γραμμή. Η μέθοδος **addMp3**, ελέχγει αν η συγκεκριμένη γραμμή τελείωνει σε '.mp3', με σκοπό να παραληφθούν όσα αρχεία δεν είναι της μορφής mp3. Αν είναι τελικά της μορφής mp3, αποθηκεύουμε στο **arrayList**, την συγκεκριμένη γραμμή του αρχείου, η οποία περιέχει και το **path** του **mp3 file.**

- **Class Main:** Δημιουργούμε ένα αντικείμενο της κλάσης **Audio,** μια μεταβλητή τύπου **InputStream** και μια μεταβλητή τύπου **M3uPlaylist,** τις οποίες και αρχικοποιούμε στο **null.** Στην συνέχεια, ελέγχουμε αν ο χρήστης έχει δώσει έως το πολύ 2 ορίσματα, αν όχι του εκτυπώνουμε ένα αντίστοιχο μήνυμα σφάλματος. Έπειτα, μέσα σε ένα **try catch**, έχουμε τις εξής περιπτώσεις: **Α)** Ελέγχουμε αν ο χρήστης έχει δώσει 1 όρισμα και αν το όρισμα που έδωσε έχει την κατάληξη ‘.mp3’, τότε καλούμε την συνάρτηση **playMp3UntilEnd. Β)** Ελέγχουμε αν ο χρήστης έχει δώσει 1 όρισμα και αν το όρισμα που έδωσε έχει την κατάληξη ‘.m3u’, τότε καλούμε την συνάρτηση **playM3uAllList. Γ)** Ελέγχουμε αν ο χρήστης έχει δώσει 2 όρισμα και αν το πρωτο όρισμα που έδωσε έχει την κατάληξη ‘.m3u’ και το δεύτερο είναι η λέξη **order**, τοτε καλούμε πάλι την συνάρτηση **playM3uAllList. Δ)** Ελέγχουμε αν ο χρήστης έχει δώσει 2 όρισμα και αν το πρωτο όρισμα που έδωσε έχει την κατάληξη ‘.m3u’ και το δεύτερο είναι η λέξη **random**, τοτε καλούμε την συνάρτηση **playM3uRandom. Ε)** Ελέγχουμε αν ο χρήστης έχει δώσει 2 όρισμα και αν το πρωτο όρισμα που έδωσε έχει την κατάληξη ‘.mp3’ και το δεύτερο είναι η λέξη **loop**, τοτε καλούμε την συνάρτηση **playMp3Loop.**